Melody harmonisation with interpolated probabilistic models

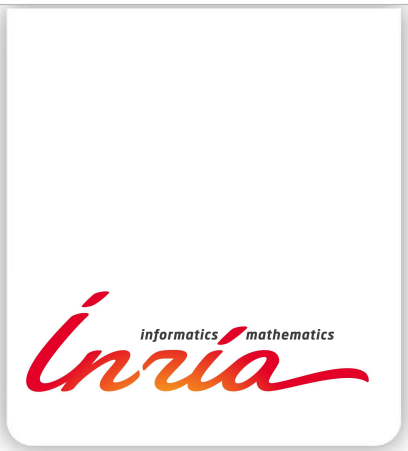
Stanislaw Raczynski, Satoru Fukayama, Emmanuel Vincent

Melody harmonisation with interpolated probabilistic models

Stanisław A. Raczyński, Satoru Fukayama, Emmanuel Vincent

RESEARCH REPORT
N° 8110
October 2012
Project-Team METISS

ISSN 0249-6399 ISRN INRIA/RR--8110--FR+ENG



Melody harmonisation with interpolated probabilistic models

Stanisław A. Raczyński, Satoru Fukayama*, Emmanuel Vincent

Project-Team METISS

Research Report n° 8110 — October 2012 — 24 pages

Abstract: Automatic melody harmonisation aims to create a matching chordal accompaniment to a given monophonic melody. Several methods have been proposed to this aim, which are generally based on musicological expertise or on unsupervised probabilistic modelling.
Among the latter category of methods, most systems use the generative hidden Markov model (HMM), in which the chords are the hidden states and the melody is the observed output. Relations to other variables, such as the tonality and scale or the metric structure, are handled by training multiple HMMs or are often simply ignored. In this paper, we propose a means of combining multiple probabilistic models of various musical variables into a versatile harmonisation system by means of model interpolation. The result is a joint model belonging to the class of discriminative models, which in recent years have proven to be capable of outperforming generative models in many tasks.
We first evaluate our models in terms of their normalized negative log-likelihood, or *cross-entropy*. We observe that log-linear interpolation offers lower cross-entropy than linear interpolation and that combining several models by means of log-linear interpolation lowers the cross-entropy compared to the best of the component models. We then perform a series of harmonisation experiments and show that the proposed log-linearly interpolated model offers higher chord root accuracy than a reference musicological rule-based harmoniser by up to 5% absolute.

Key-words: Dynamic Bayesian Networks, melody harmonisation, model interpolation

* The University of Tokyo

RESEARCH CENTRE
RENNES – BRETAGNE ATLANTIQUE

Campus universitaire de Beaulieu
35042 Rennes Cedex

Harmonisation de mélodies par interpolation de modèles probabilistes

Résumé : L'harmonisation automatique de mélodies vise à créer une suite d'accords accompagnant une mélodie donnée. Plusieurs méthodes ont été proposées dans ce but, généralement basées sur des règles musicologiques expertes ou sur une modélisation probabiliste non supervisée.

Parmi cette dernière catégorie de méthodes, la plupart utilisent un modèle de Markov caché (MMC) génératif, dont les états cachés sont les accords et l'observation la mélodie. Les dépendances aux autres variables telles que la tonalité ou la structure métrique sont modélisées par des MMCs multiples ou simplement ignorées. Dans ce papier, nous proposons un moyen de combiner plusieurs modèles probabilistes de différentes variables musicales par le biais d'une interpolation de modèles. Cela aboutit à un modèle combiné appartenant à la catégorie des modèles discriminants, dont il a été démontré ces dernières années qu'ils dépassent la performance des modèles génératifs pour de nombreuses tâches.

Nous évaluons d'abord nos modèles en terme de l'opposé de leur log-vraisemblance normalisée, ou *entropie croisée*. Nous observons que l'interpolation log-linéaire diminue l'entropie croisée par rapport à l'interpolation linéaire et que la combinaison de plusieurs modèles par interpolation log-linéaire diminue l'entropie croisée par rapport au meilleur modèle individuel. Nous effectuons ensuite un ensemble d'expériences d'harmonisation et montrons que le modèle par interpolation log-linéaire proposé améliore la précision d'estimation de la fondamentale des accords de 5% dans l'absolu par rapport un algorithme d'harmonisation de référence basé sur des règles musicologiques expertes.

Mots-clés : Réseaux bayésiens dynamiques, harmonisation de mélodies, interpolation de modèles probabilistes

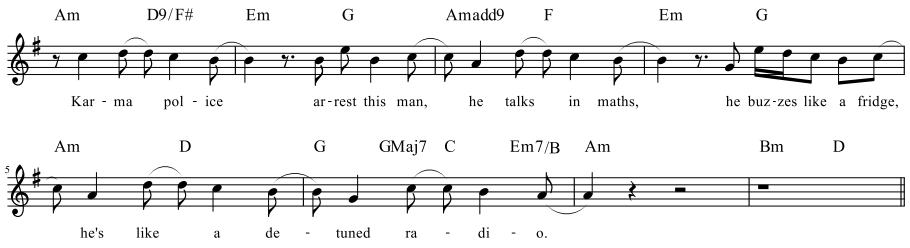


Figure 1: Example harmonisation of the lead melody from Radiohead's *Karma Police*.

1 Introduction

Automatic melody harmonisation is the process of determining the most musically suitable chordal accompaniment to a given monophonic melody. It is an important part of musical composition and so it is a common exercise in all music composition classes, where it typically involves determining a *four-part harmony*, *i.e.*, the movement of four voices, namely: soprano, alto, tenor and bass. The task can be either to find the harmonization for a given melody performed by the soprano voice, or to find the three other voices for a given bass line (*unfigured bass*), often with some chordal information given (*figured bass*). In this work, however, we focus on harmonisation in the more narrow senseof generating a sequence of background chords matching a given melody, which can be played on supporting instruments, *e.g.*, on a guitar or a piano. It is simpler than four-part harmonisation because one does not need to determine the exact movements of the voices—it does not include inversions, added and removed tones, *etc.*, but only the root pitch (C, C♯, *etc.*) and the chord type (major, minor, *etc.*). The melody together with the chord labels are typically referred to as *lead sheets*, an example of which is shown in Fig. 1.

Harmonisation is a necessary step in most algorithmic composition methods, which typically involve generating a melodic line first, and afterwards supplementing it with an accompaniment. For example, in their *Orpheus* automatic composition system, Fukayama *et al.* compose a melody based on constraints resulting from the tonal accent in the lyrics and from basic musicological rules, and then compose the chordal accompaniment using a collection of accompaniment patterns [12]. Harmonisation has also been explored as an easy way to create polyphonic ring-tones from simple melodies in the *i-Ring* ring-tone harmonisation system from [20]. Furthermore, automatic harmonisation has recently received significant commercial interest as an easy way for non-musicians to create well-sounding music based on simple melodies. The most well-known implementations are the *MySong* software developed in cooperation with Microsoft [31] and the commercial software package *Band-in-a-Box* (BIAB) [14]. Both are designed as tools for non-professional musicians, or even non-musicians, to create songs with instrumental accompaniment by singing a melody into a

microphone. All of the above methods perform a lead sheet-like harmonisation.

The history of automatic harmonisation is much older, however. Some of the earliest attempts at harmonisation were made by Steels, who proposed a heuristic search approach in [32], and by Ebcioğlu, who proposed a rule-based system targeting Bach's chorales [9, 10]. Another rule-based harmoniser based on musicological expertise—called *Harmonic Analyzer*—was more recently developed by Temperley and Sleator in [33, 34]. Other harmonisation methods have also been explored. Phon-Amnuaisuk and Wiggins developed a prototype system based on genetic algorithms with knowledge-rich structures in [25]. Constraint satisfaction systems were investigated by Pachet and Roy in [22] and sequential neural networks by Gang *et al.* in [13]. Neural networks were also used by Cunha and Ramalho in their hybrid neural-musicological real-time chord prediction system in [7]. Another hybrid harmonisation system that combines Hidden Markov Models (HMMs) with a set of heuristic rules for rapid training was proposed by Chuan and Chew in [5]. Among these, only [34], [13], [7] and [5] are lead sheet-like harmonisation systems.

Among the most flexible systems are those based on unsupervised probabilistic modelling, which typically utilise HMMs. Single-HMM approaches include the *MySong* software and an implementation based on *MySong* from [4], as well as the *i-Ring* ring-tone harmonisation system from [20]. A slightly more complicated four-part harmonisation approach using a dual HMM (one for harmonisation and another one for ornamentation) was proposed by Allan and Williams in [2]. Later, in [23] Paiment *et al.* proposed a very sophisticated, multi-level graphical model for modelling chord movement with respect to the melody, which is capable of modelling long-term relationships between chords, as opposed to HMMs that are only capable of modelling short-term dependencies. However, being a non-dynamic graphical network, their model is limited to fixed-length songs (of exactly 16 measures). By contrast with musicological rule-based methods, which require careful formulation and application of harmonisation rules for particular genres (*e.g.*, classical or jazz), these probabilistic methods aim to automatically infer those rules from a corpus of example data and are therefore applicable to all genres, even when musicological expertise is not available.

All of the above probabilistic harmonisation systems are however limited to modelling the relation between a single hidden layer (chord sequence) and a single observed layer (melody), without any explicit mechanism to include models of other relevant musical quantities, such as the key and the current tonal centre, the rhythm and the musical accent, or the genre, period and the composer. One can use multiple HMMs corresponding to, *e.g.*, different genres or keys as in [31], but with many variables this approach quickly suffers from over-fitting. In this paper we propose to build versatile chord models by interpolating between a collection of simpler *sub-models* using linear or log-linear interpolation.

This paper is organised as follows. Section 2 explains the proposed modelling and training approach and Section 3 gives details about the particular sub-models used in our experiments. The experimental set-up and the results are described in Section 4. Finally, the conclusion is given in Section 5.

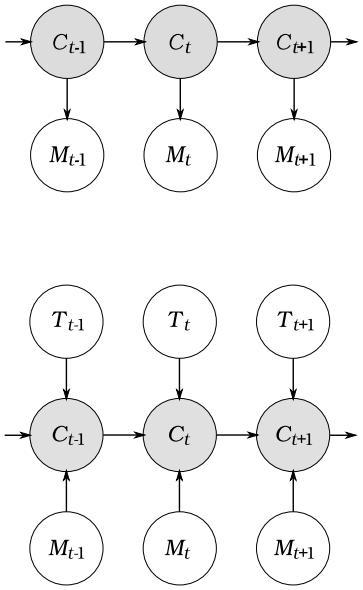


Figure 2: A typical HMM for melody harmonisation (top) compared to the proposed model (bottom).

2 General approach

When figuring out the accompaniment, one needs to keep in mind the basic rules of tonal music: the tonality (everything in tonal music happens with respect to the tonic), the chord progressions (certain chord progressions are more natural and pleasant, *e.g.*, progressions corresponding to the circle of fifths, progressions descending by thirds and common cadences), and of course harmonic compatibility with the melody. In the *generative*, HMM-based systems, the current chord is the underlying state C_t, while the observation is the melody M_t. Chord progression is modelled with a Markov chain $\mathrm{P}(C_t|C_{t-1})$ and the observed melody by a multinomial distribution conditioned on the system's state $\mathrm{P}(M_t|C_t)$ (see the top of Fig. 2).

2.1 Model structure

We propose a more flexible way of developing probabilistic harmonisation models, in which the time-varying tonality T_t, as well as other musical variables can be explicitly taken into account. We propose a *discriminative* model, in which the chords are modelled conditionally to all other variables (see bottom

of Fig. 2):

$$\mathrm{P}(C_t|\mathbf{C}_{1:t-1}, \mathbf{X}_{1:t}), \tag{1}$$

where $1:t$ denotes a range of time indices from the first to the current time frame and $\mathbf{X}$ is a set of other musical variables, such as the melody, the tonal centre, the metrical accent, the style or genre, *etc.* Discriminative models have been found to outperform generative models in many fields [16], such as speech recognition [28, 35], machine translation [21], text classification [29] or genomics [15].

This conditional multinomial distribution has too many parameters to be used in practice, hence we approximate it by interpolating between multiple sub-models P_i involving a different subset of conditioning variables $\mathbf{A}_{i,t} \subset \{C_{1:t-1}, \mathbf{X}_{1:t}\}$. The interpolation can be linear,

$$\mathrm{P}(C_t|\mathbf{C}_{1:t-1}, \mathbf{X}_{1:t}) = \sum_{i=1}^{I} a_i \mathrm{P}_i(C_t|\mathbf{A}_{i,t}), \tag{2}$$

with

$$\sum_{i=0}^{I} a_i = 1, \tag{3}$$

or log-linear,

$$\mathrm{P}(C_t|\mathbf{C}_{1:t-1}, \mathbf{X}_{1:t}) = Z^{-1} \prod_{i=1}^{I} \mathrm{P}_i(C_t|\mathbf{A}_{i,t})^{b_i}, \tag{4}$$

where I is the number of sub-models, $a_i \geq 0$ and $b_i \geq 0$ for $i = 1, \ldots, I$ are the interpolation coefficients and

$$Z = \sum_{C_t} \prod_{i=1}^{I} \mathrm{P}_i(C_t|\mathbf{A}_{i,t})^{b_i} \tag{5}$$

is a normalizing factor depending on $\mathbf{A}_{i,t}$. For example, we will consider in the following: $\mathbf{A}_{1,t} = \{C_{t-1}\}$, $\mathbf{A}_{2,t} = \{T_t\}$ and $\mathbf{A}_{3,t} = \{M_t\}$.

Linear [17] and log-linear [19] interpolation have been previously used in the context of natural (spoken) language modelling to combine models with different temporal spans (n-grams with different values of n). Here we have generalized this approach to interpolate between sub-models conditioned on different musical variables.

2.2 Smoothing

Although the sub-models P_i now have fewer parameters, over-fitting issues may still arise due to data sparsity, so the above equations are not directly usable. In order to address these issues, each of the sub-models must be smoothed [36]. In this study we perform smoothing by combining each sub-model with the prior

chord distribution and a uniform distribution. In the case of linear interpolation, the smoothing is applied to the interpolated model:

$$\mathrm{P}(C_t|\mathbf{C}_{1:t-1},\mathbf{X}_{1:t}) = \sum_{i=1}^{I} a_i \mathrm{P}_i(C_t|\mathbf{A}_{i,t}) + \alpha \mathrm{P}(C_t) + \beta, \tag{6}$$

with

$$\alpha + \beta + \sum_{i=1}^{I} a_i = 1. \tag{7}$$

In the case of the log-linear interpolation, each model is smoothed separately before combining them:

$$\mathrm{P}(C_t|\mathbf{C}_{1:t-1},\mathbf{X}_{1:t}) = Z^{-1} \prod_{i=1}^{I} \left(\gamma_i \mathrm{P}_i(C_t|\mathbf{A}_{i,t}) + \delta_i \mathrm{P}(C_t) + \epsilon_i\right)^{b_i}, \tag{8}$$

with

$$\gamma_i + \delta_i + \epsilon_i = 1 \tag{9}$$

for all i and

$$Z = \sum_{C_t} \prod_{i=1}^{I} \left(\gamma_i \mathrm{P}_i(C_t|\mathbf{A}_{i,t}) + \delta_i \mathrm{P}(C_t) + \epsilon_i\right)^{b_i}. \tag{10}$$

2.3 Training

The proposed models are trained on two disjoint sets of example data called *training set* and *validation set*. The sub-models P_i and the prior distribution $\mathrm{P}(C_t)$ are first trained in the maximum likelihood (ML) sense on the training set by counting occurrences [36]. The interpolation coefficients a_i or b_i and the smoothing coefficients α and β or γ_i, δ_i and ϵ_i are then jointly trained on the validation set according to one of two possible training objectives.

Classical generative training is achieved by estimating the interpolation and smoothing coefficients in the ML sense on the validation set. Because the log-likelihood is convex [19], any optimization algorithm can be used. In the following, we have used a non-negatively constrained limited-memory Broyden-Fletcher-Goldfarb-Shanno (BFGS) method (a quasi-Newton optimisation), built into the GNU R environment as the `optim()` function [26].

Even though the likelihood is a common evaluation measure for statistical models, higher likelihood does not always translate into better performance for the considered application. For example, it is well known in the field of automatic speech recognition that the likelihood of language models can sometimes be improved without effect on the word error rate [3]. For that reason, we alternatively propose to perform discriminative training by estimating the interpolation and smoothing coefficients so as to maximize the chord root note accuracy on the validation set, which is the main evaluation metric used in the harmonisation experiments in Subsection 4.2. Because this metric is not

differentiable, gradient-based methods cannot be used so we have used the following multi-step brute-force search. First, all smoothing coefficients are fixed to $\alpha_i = 0.1$ and $\beta_i = 0.5$ (values chosen experimentally) and the interpolation coefficients are optimised by testing all combinations of values between 0 and 1 in 0.1 steps (11 distinct values). Then, the smoothing coefficient pairs are optimised separately and sequentially in the same range. Finally, the interpolation coefficients are fine-tuned around the original optimum (±20%, 11 values) using the newly trained smoothing coefficients.

3 Sub-models

As a proof of concept, we have developed three sub-models that model the three most important aspects of chords: chord progressions, relation to the tonality and relation to the melody:

$$\mathrm{P}_1 = \mathrm{P}(C_t|C_{t-1}), \tag{11}$$

$$\mathrm{P}_2 = \mathrm{P}(C_t|T_t), \tag{12}$$

$$\mathrm{P}_3 = \mathrm{P}(C_t|M_t). \tag{13}$$

To train these three sub-models, we have used a collection of around 2000 lead sheets encoded in the MusicXML format that are freely available on the Wikifonia web page [11]. We converted each lead sheet into three sequences of symbols representing tonality, melody and chords. In order to do so, we first partitioned the input into regular time frames of length 1/3, 1/2, 1, 2, 4, 8 or 16 beats. For every frame, we defined the melody variable M_t as the unordered list of pitch classes of the 12-tone chromatic scale appearing in the melody, which can take $2^{12} = 4096$ distinct values. This is similar to the melody encoding in [4] and [27]. The tonality T_t was encoded as one of 24 different key labels resulting from the combination of 12 tonics (C, C♯, D, D♯, E, F, F♯, G, G♯, A, A♯, B) and 2 modes (major or minor). The chord C_t was labelled by one of 13 root pitch classes (C, C♯, D, D♯, E, F, F♯, G, G♯, A, A♯, B or "none" for non-chords) and one of 27 chord types (major, minor, dominant, diminished, half-diminished, augmented, power, suspended-second, suspended-fourth, major-sixth, minor-sixth, major-seventh, minor-seventh, dominant-seventh, diminished-seventh, augmented-seventh, major-ninth, minor-ninth, dominant-ninth, augmented-ninth, minor-eleventh, dominant-eleventh, major-minor, minor-major, major-thirteenth, dominant-thirteenth or "none" for non-chords), resulting in $N = 351$ distinct chord labels in total. The chord C_0 before the beginning of the song was assumed by convention to be "none". In the case of a time frame containing more than one key or chord, the longest lasting key and chord labels within that frame were selected.

The distribution of tonalities in the training set is shown in Fig. 3. The C-major key appears to be dominant, which will have an impact on the design of the models in order not to bias them toward that particular tonality, as explained in Subsection 3.2.

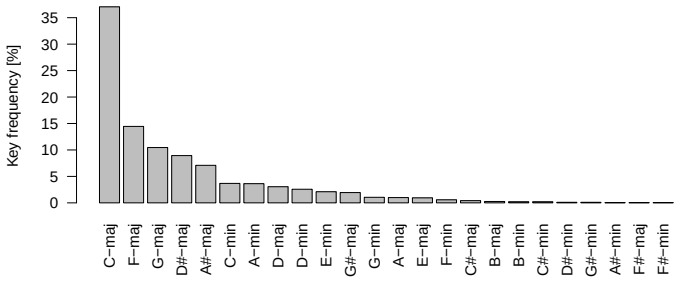


Figure 3: Histogram of the key labels in the training dataset.

3.1 Chord prior

The chord prior $\mathrm{P}(C_t)$ is used for smoothing and as a reference model in the evaluation. The prior chord distribution trained on the training dataset is presented in Fig. 4. Note that the major, dominant, minor, minor- and major-seventh chords make up for the vast majority of the chords in the dataset. Also, due to the dominance of the C-major key in the training set, the pitch classes C, F and G have visibly higher probabilities than the other root pitch classes.

3.2 Chord bigram model

The chord progression model is built under the Markov assumption, resulting in a bigram model $\mathrm{P}_1(C_t|C_{t-1})$. Although longer chordal dependencies can perform better, this has already been studied by others [23, 30] and is not the focus of this paper. In order to avoid problems with data sparsity, the model was trained with state tying: probabilities of all *relative* chord transitions were tied together, so for example the probability of transition from C-major to G-minor (7 semitones) is identical to that of transition from G-major to D-minor (also 7 semitones). This is motivated by the observation that in tonal music songs can be freely transposed between all keys without any loss of musical correctness [24], and by the dominance of the C-major key in the training set (see Fig. 3), which would otherwise result in a biased chord distribution towards the common chords of that key.

The resulting conditional chord distribution can be observed in Fig. 5. The distribution for a 1-beat analysis frame (top of Fig. 5) is very concentrated towards the previous chord (G-major to G-major transition), because chords typically last for at least few beats. On the other hand, using a 16-beat analysis frame makes the bigram probabilities being more evenly distributed, though still dominated by the transition to the same chord.

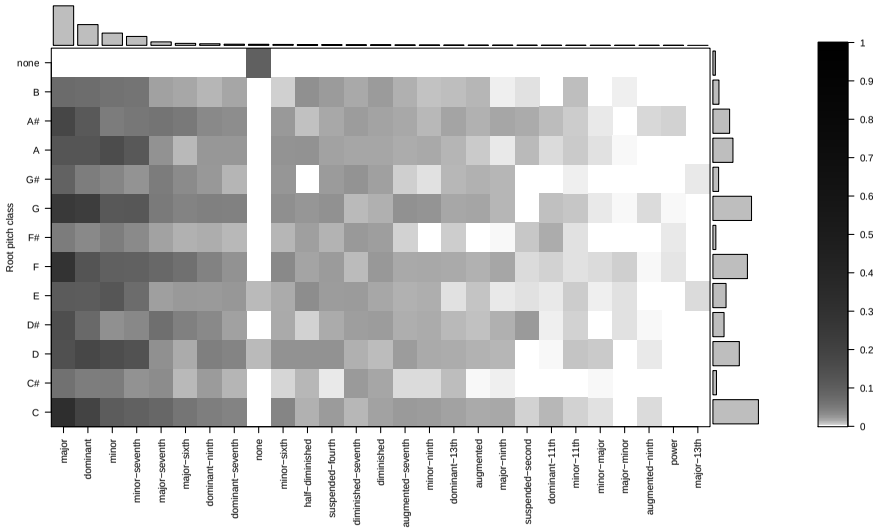


Figure 4: Distribution of chords $\mathrm{P}(C_t)$ for 1-beat frames. On top: distribution of chord types; on the right: distribution of chord roots. Chord types are sorted by their occurrence frequency.

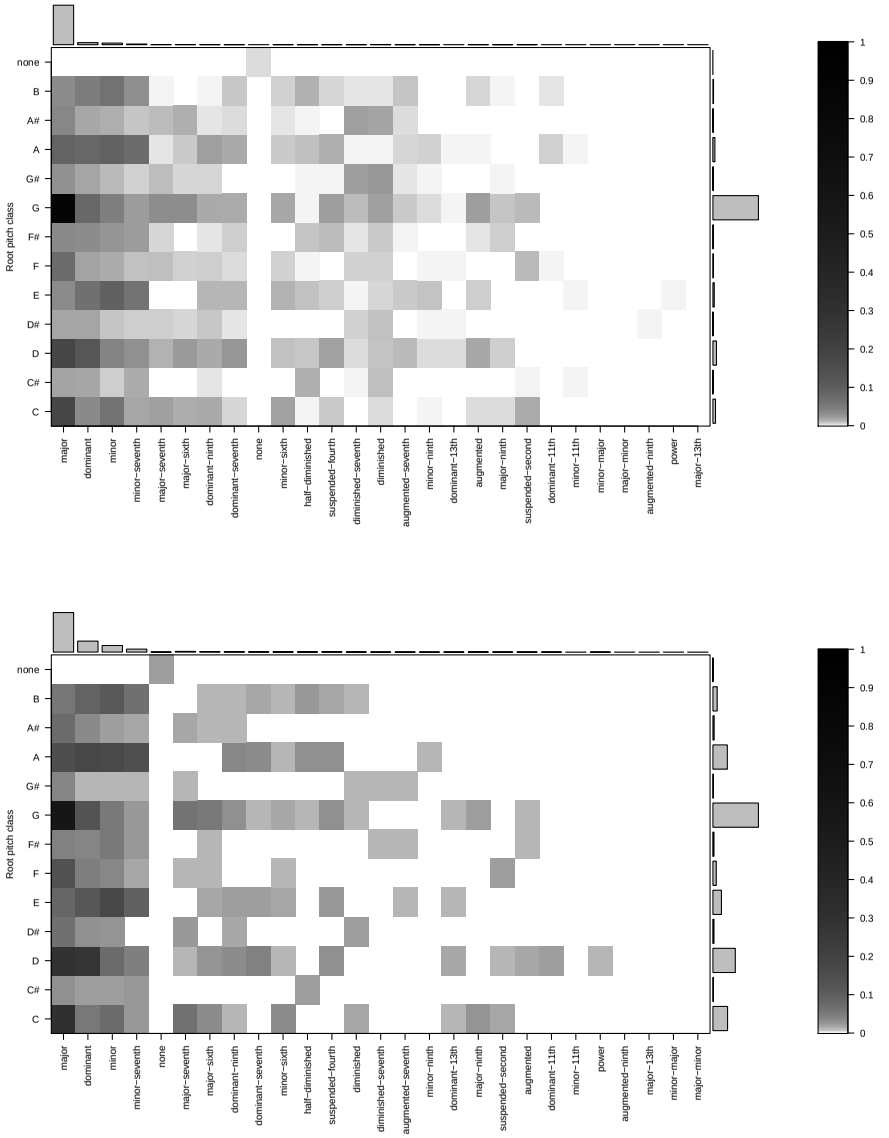


Figure 5: Conditional distribution of chords $\mathrm{P}_1(C_t|C_{t-1})$ (chord transition probability) for 1-beat frames (top) and 16-beat frames (bottom), plotted for the previous chord C_{t-1} = G-major. The plot is accompanied with linear-scale bar plots of chord root and chord type distributions on the sides.

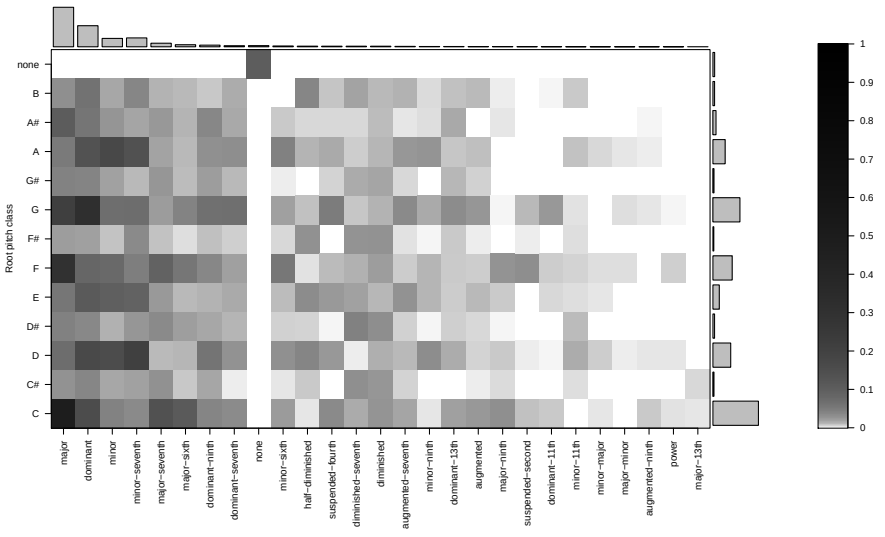


Figure 6: Conditional distribution of chords $\mathrm{P}_1(C_t|T_t)$ for 1-beat frames, plotted for the tonality of T_t = C-maj. The plot is accompanied with linear-scale bar plots of chord root and chord type distributions on the sides.

3.3 Tonality model

In the tonality model $\mathrm{P}_2(C_t|T_t)$, for the same reasons as explained in Subsection 3.2, the chords corresponding to the same scale degree in different keys were tied together. In other words, observing, *e.g.*, a dominant major chord in one key increases the probability of dominant major chords in all keys. The resulting tonality model distribution is depicted in Fig. 6. Notice that knowing the current tonality to be, in this case, T_t = C-major increases the dominance of the most common degrees: the dominant (V), subdominant (IV), supertonic (ii), but mostly the tonic (I).

3.4 Melody model

Finally, for the same reasons again, state tying was used for the melody model $\mathrm{P}_3(C_t|M_t)$ as well. Note patterns with the same content relative to the chord root were given identical probabilities, *e.g.*, the unordered note combination (C,G) in the chord of C-major is equally probable as the note combination (D♯,A♯) in the chord of D♯-major. The resulting melody model distribution is shown in Fig. 7. Note that having more melodic information, *i.e.*, more notes in a frame (here C, E and G in one frame in the bottom part of the plot), makes the chord distribution significantly sparser. This means that a system with longer frames will have a more informative melody model, because a single

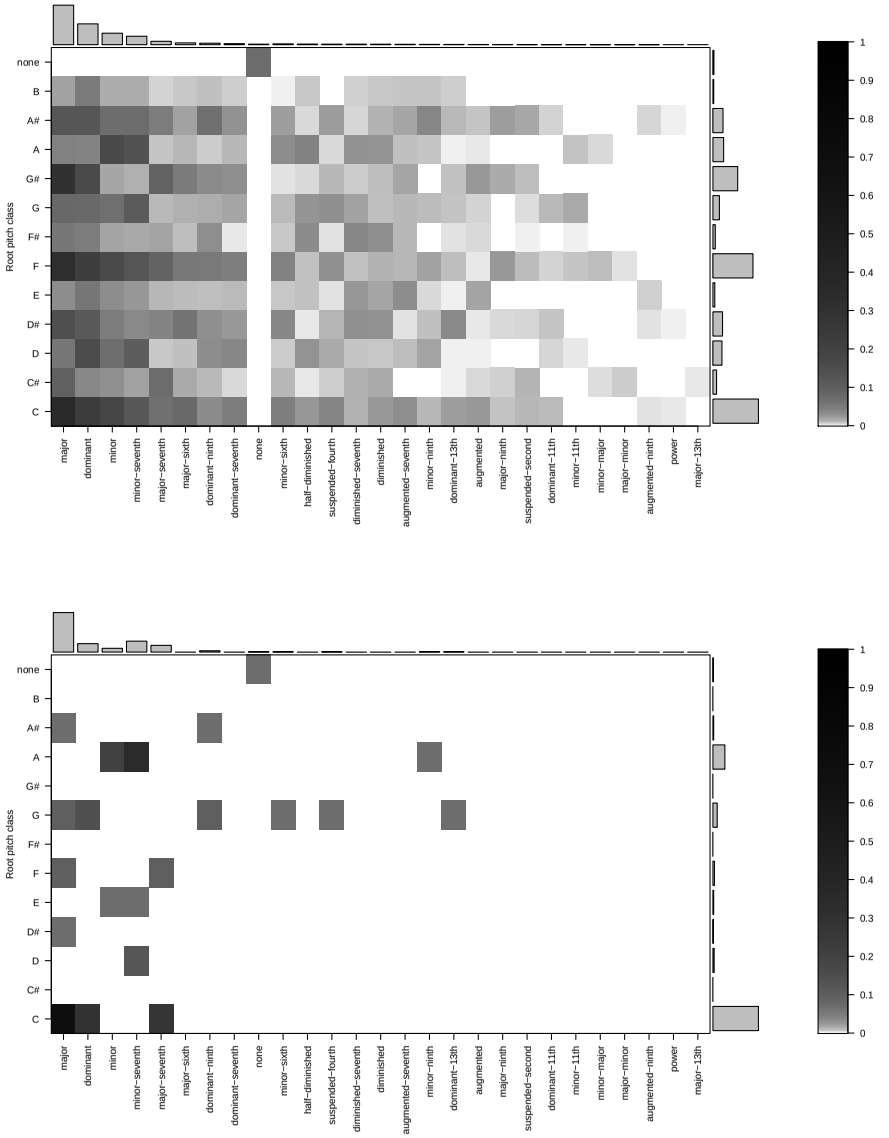


Figure 7: Conditional distribution of chords $\mathrm{P}_3(C_t|M_t)$ for 1-beat frames, plotted for a single C note (top) and for a time frame containing three notes: C, E and G (bottom). Each plot is accompanied with linear-scale bar plots of chord root and chord type distributions on the sides.

frame will contain more melody notes.

3.5 Interpolation coefficients

The values of the generatively trained interpolation coefficients a_i and b_i are presented in Fig. 8. The coefficients for linear and log-linear interpolation follow a similar trend: the bigram model is given a progressively lower weight as the frame length increases, while the melody model behaves in an opposite manner. Indeed, for large frames the bigram model becomes less informative due to the lack of need of modelling the chord duration, while the melody model becomes more informative due to the larger melodic context.

4 Evaluation

Due to a large range of goals in the existing literature, we can observe a variety of ways of evaluating harmonisation algorithms: through theoretic evaluation of the modelling power by means of cross-entropies [2, 23], through comparison of generated chord sequences with ground-truth chord annotations [4, 5], through comparison of single predicted chords with ground-truth [7], or through subjective listening tests [31, 20]. Because of the novelty of the proposed solutions, as well as that of the field of automatic harmonisation itself, many papers did not offer any evaluation [25, 9, 22, 13]. In this paper, we have chosen to perform two complementary evaluations: we first use the theoretic cross-entropy-based evaluation as in [2, 23] as a convenient way to validate our interpolation-based approach and we then perform an objective evaluation of the generated chord sequences in the same manner as in [4]. The code of our algorithm is available online at `http://versamus.inria.fr/software-and-data/harmonization.tbz2`.

Out of the 2000 Wikifonia lead sheets, 100 lead sheets were used as a test set, 100 as a validation set, and the rest were used as a training set.

4.1 Cross-entropy

An efficient way of determining the modelling power of a model is to compute the normalized negative log-likelihood, or *cross-entropy* [18]. For base-2 logarithms, it can be interpreted as the average number of bits (b) required to encode a single chord symbol (Shannon's optimal code length). So, naturally, the smaller the value, the higher the prediction power. The cross-entropy is calculated from the test chord, melody and tonality sequences $\mathbf{C}$, $\mathbf{M}$ and $\mathbf{T}$ as

$$\mathrm{H}(\mathbf{C}) = -\frac{1}{T}\log_2 \mathrm{P}(\mathbf{C}|\mathbf{M},\mathbf{T},\Lambda) = -\frac{1}{T}\sum_{t=1}^{T}\log_2 \mathrm{P}(C_t|\mathbf{C}_{1:t-1},\mathbf{X}_{1:t}) \qquad (14)$$

where Λ denotes the model parameters and T is the number of frames in the test set. The cross-entropy is upper bounded by the cross-entropy of the non-

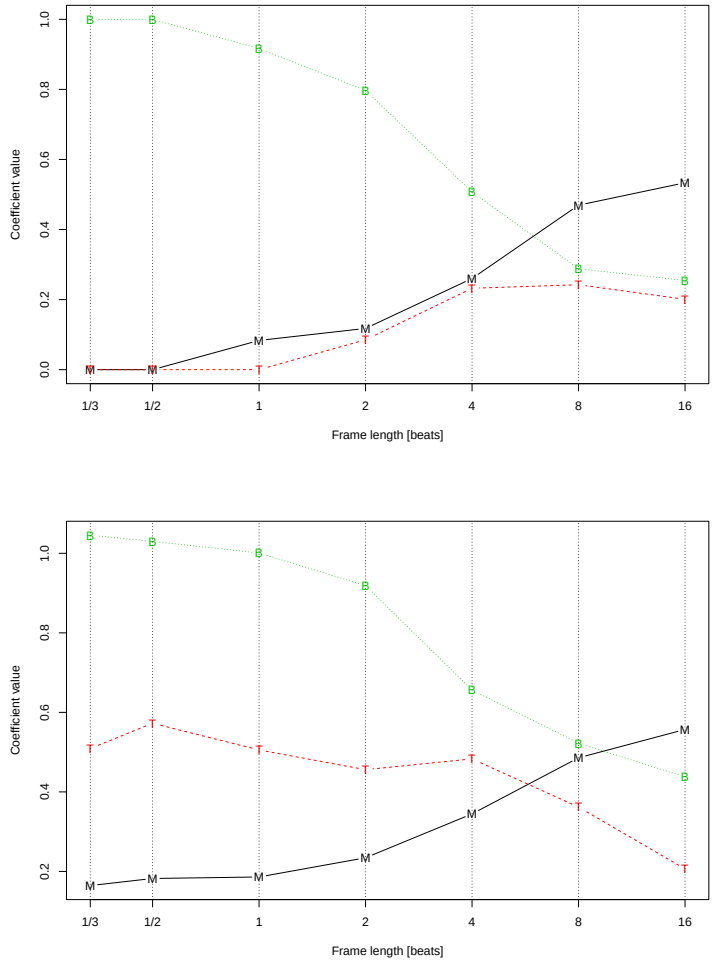


Figure 8: Values of the generatively trained interpolation coefficients for linear (top) and log-linear (bottom) interpolation.

informative uniform chord prior, which in our case is equal to

$$\mathrm{H}_U = -\log_2 \frac{1}{N} = 8.46 \text{ b}, \tag{15}$$

where $N = 351$ is the number of distinct chord labels. The cross-entropies obtained for different frame lengths (from 1/3 to 16 beats) and different generatively trained log-linear combinations of the bigram (B), tonality (T) and melody (M) models are plotted in the upper part of Fig. 9. We can observe that the predicting power of the melody (M) and tonality (T) models, although better than the prior alone, is poor for small frame lengths (about 5 bits/frame), but improves by about 0.5 bits/frame as the frame length increases. This is logical, as both the tonality and the chord are musical quantities that depend on a much wider context than a single melody note. However, using large frame lengths limits the temporal precision of the chord estimation, since chords can typically change on any beat (though typically on the down beat). We also observe the benefit of model interpolation: the combined melody and tonality (M+T) model is better than either of the two models alone, and the combined melody and tonality and bigram (M+T+B) model is better the each of these three models alone and than the M+T combination. For 2-beat frames, the latter improvement is equal to 0.37 bit/frame (11%).

Still in the upper part of Fig. 9, we can see that the cross-entropy decreases monotonically with decreasing frame length for those models that include the bigram chord progression model, namely B and M+T+B. In fact, it would decrease asymptotically to zero for infinitesimal frame lengths, because predicting the next chord given the current one would be getting easier: one would simply have to predict the same chord and that prediction would be increasingly correct. Therefore we found it useful to complement this plot with a plot of cross-entropies normalised per beat instead of per frame in the bottom part of Fig. 9. Per-beat cross-entropies are proportional to the total amount of information in the entire dataset, given the model (because the number of beats in the dataset is fixed, as opposed to the number of frames). From that plot we can see that we actually get more informative bigram models for larger frame lengths. On the other hand, this evaluation measure is biased towards larger frame lengths since it is bounded by the per-beat cross-entropy of the uniform chord distribution $H_U/B = (\log_2 N)/B$, where B is the number of beats per frame, which decreases asymptotically towards zero as B increases.

In the light of the above discussion we conclude that the best frame length for harmonisation is in the range of few beats, where the cross-entropy is between 2 bits/frame for 1-beat frames and 4 bits/frame for 8-beat frames.

Fig. 10 shows the reduction of cross-entropy achieved by log-linear interpolation with respect to linear interpolation ($\mathrm{H}_{\mathrm{lin}} - \mathrm{H}_{\mathrm{loglin}}$). Although log-linear interpolation is more time consuming (due to the need for re-normalisation by Z), it offers significantly higher modelling power when several models are combined: up to 0.32 bit/frame (10%) for the M+T model and up to 0.26 bit/frame (6.5%) for the M+T+B model.

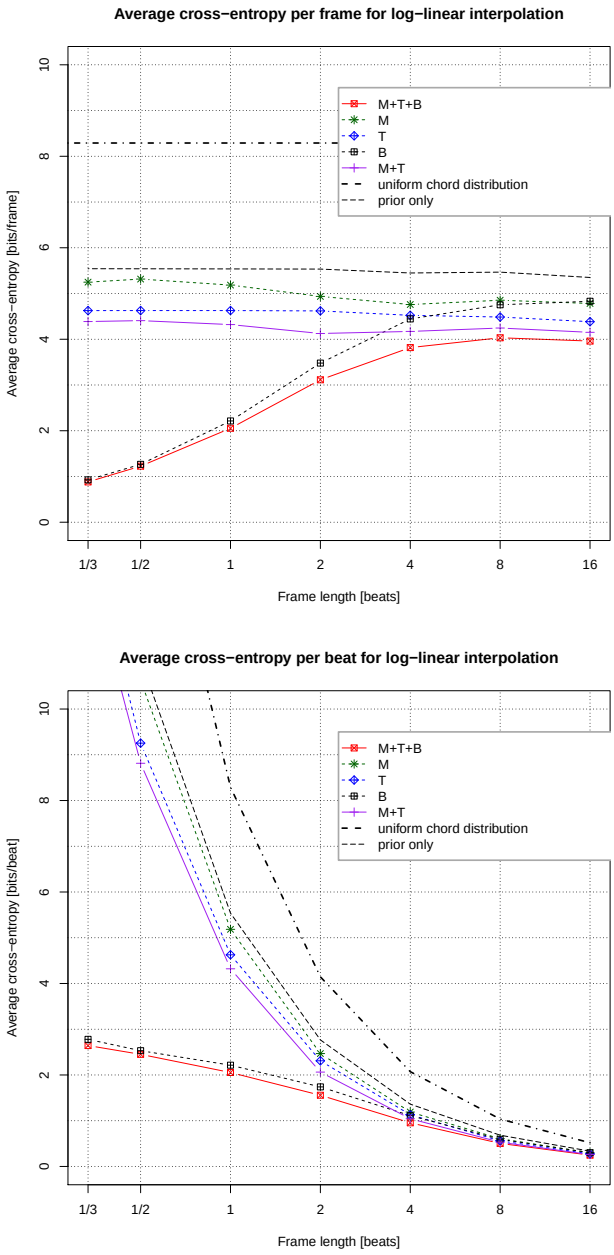


Figure 9: Cross-entropies calculated for the test dataset, normalised per time frame (top) and per beat (bottom). M stands for the melody model, T for the tonality model and B for the bigram chord progression model. Models were combined using log-linear interpolation.

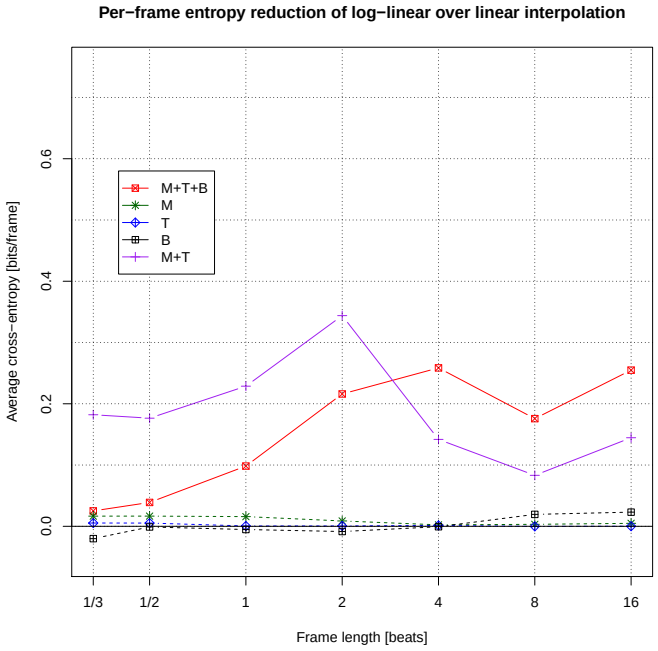


Figure 10: Cross-entropy reduction of log-linear over linear interpolation.

4.2 Harmonisation

In a second experiment we compare the chord sequence generated by the full discriminatively trained model M+T+B with the ground truth chord labels in the test lead sheet files. In this experiment, the timing of the ground truth chord sequence is preserved and does not depend on the chosen frame length. Chords were estimated via a Viterbi-like algorithm, which finds the most likely sequence of chords given the melody and tonality. The tonality was assumed to be known, because often several tonal interpretations are possible [1] and we want the resulting chord sequences to be comparable to the ground truth. Similarly to the MIREX competition [8, 6], the estimated chord types were subsequently clustered into a smaller number of triads (major, minor, augmented, diminished, suspended second or suspended fourth). We compare the chords either in terms of their root note only or in terms of their root note and their triad chord type using two alternative root accuracy measures: a binary one and a weighted one. The latter uses the following weights: 1 for correct root pitch class estimation, 0.5 for 5 or 7 semitone errors (perfect fourth or fifth), 0.3 for 3, 4, 8 or 9 semitone errors (minor or major third and minor or major sixth), 0.1 for 2 or 10 semitone errors (major second or minor seventh) and 0 for other errors.

For comparison, we have used the results of the state-of-the-art rule-based lead sheet-like harmonisation system of Temperley and Sleator, which is freely available on-line at [34]. The lead sheets were converted to the input format of that algorithm with a time precision ("BaseUnit") of a dotted sixty-fourth note and the metrical structure was generated based on the time signatures and the upbeat durations extracted from the MusicXML files. Note that this algorithm estimates only the chord roots, not the chord types.

The results for different frame lengths are plotted in Figs 11, 12 and 13. For all evaluation metrics, the log-linearly interpolated models offer better accuracy than the linear ones and the best results are most often achieved for a frame length of 2 beats. For shorter frame lengths the melody provides less information, while for longer frame lengths the temporal resolution of the generated chord sequence becomes too coarse. For that frame length, the proposed algorithm with log-linear interpolation outperforms the reference algorithm by 5.5% absolute (17% relative) in terms of root note accuracy and by 4.1% absolute in terms of weighted root note accuracy.

5 Conclusion

In this paper we have presented a novel method of building versatile statistical models of chords for harmonisation by joining multiple simpler sub-models by means of linear or log-linear interpolation. To test this idea, we have trained and combined in this way three sub-models: the tonality, the melody and the chord bigram model. We have evaluated the resulting interpolated models in terms of their cross-entropy and observed that log-linear interpolation yields a model whose cross-entropy is lower than the best of the component models

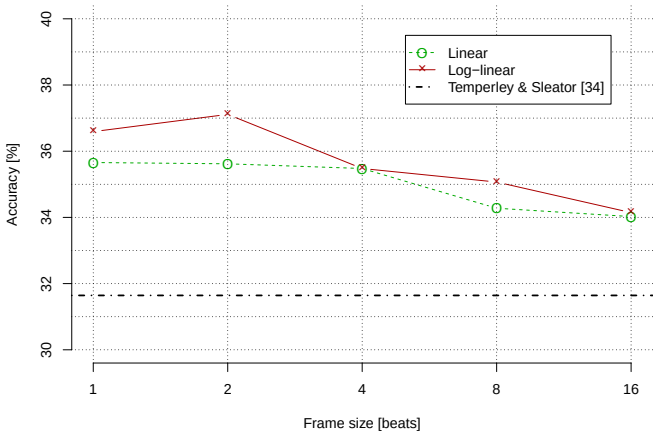


Figure 11: Root note accuracy obtained for different model frame lengths.

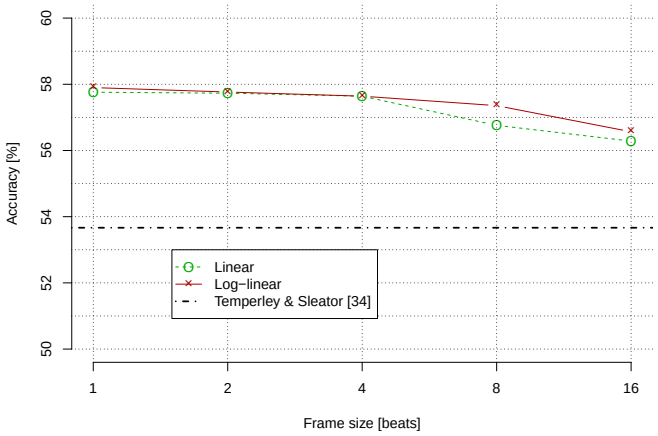


Figure 12: Weighted root note accuracy obtained for different model frame lengths.

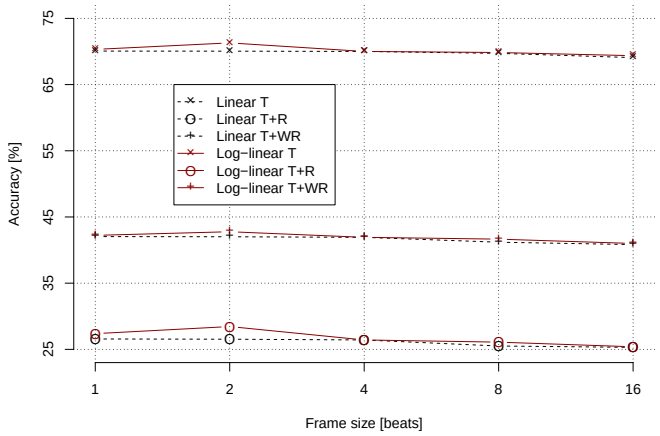


Figure 13: Triad accuracies obtained for different model frame lengths: triad chord type alone (T), chord type and root note (T+R), and chord type and weighted root note (T+WR).

and also better than tha achieved by linear interpolation. We have then performed a series of harmonisation experiments, where we have observed that the proposed log-linearly interpolated model offers higher root chord accuracy than the reference rule-based harmoniser from [33] by up to 5% absolute.

In future work, a larger number of more complex sub-models could be investigated for further improvement in terms of chord accuracy. Subjective listening tests could also be used to analyse the quality of the harmonisations in more details. Finally, the model interpolation methodology could be applied to other music information retrieval tasks that would potentially benefit from modelling several musical aspects simultaneously.

Acknowledgment

This work was supported by INRIA under the Associate Team Program VERSAMUS (http://versamus.inria.fr/).

References

[1] R. Aiello and J.A. Sloboda. *Musical perceptions*. Oxford University Press New York and Oxford, 1994.

[2] M. Allan and C.K.I. Williams. Harmonising chorales by probabilistic inference. *Advances in Neural Information Processing Systems*, 17:25–32, 2005.

[3] S. Chen, D. Beeferman, and R. Rosenfeld. Evaluation metrics for language models. In *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, 1998.

[4] C.H. Chuan. A comparison of statistical and rule-based models for style-specific harmonization. In *Proc. 12^{th} International Society for Music Information Retrieval Conference (ISMIR)*, pages 221–226, 2011.

[5] C.H. Chuan and E. Chew. A hybrid system for automatic generation of style-specific accompaniment. In *Proc. 4^{th} International Joint Workshop on Computational Creativity*, 2007.

[6] MIREX community. Music Information Retrieval Evaluation eXchange. `http://www.music-ir.org/mirex/wiki/MIREX_HOME`, August 2012.

[7] U.S. Cunha and G. Ramalho. An intelligent hybrid model for chord prediction. *Organised Sound*, 4(2):115–119, 1999.

[8] J. Downie, A. Ehmann, M. Bay, and M. Jones. The music information retrieval evaluation exchange: Some observations and insights. In *Advances in Music Information Retrieval*, chapter 6, pages 93–115. Springer, 2010.

[9] K. Ebcioğlu. An expert system for chorale harmonization. In *Proc. National Conference in Artificial Intelligence (AAAI)*, 1986.

[10] K. Ebcioğlu. An expert system for harmonizing four-part chorales. *Computer Music Journal*, 12(3):43–51, 1988.

[11] Wikifonia Foundation. Wikifonia. `http://www.wikifonia.org/`, August 2012.

[12] S. Fukayama, K. Nakatsuma, S. Sako, T. Nishimoto, and S. Sagayama. Automatic song composition from the lyrics exploiting prosody of the Japanese language. In *Proc. 7^{th} Sound and Music Computing Conference (SMC)*, pages 299–302, 2010.

[13] D. Gang, D. Lehman, and N. Wagner. Tuning a neural network for harmonizing melodies in real-time. In *Proc. International Computer Music Conference (ICMC)*, 1998.

[14] PG Music Inc. Band-in-a-box. `http://www.pgmusic.com/`, August 2012.

[15] T. Jaakkola, M. Diekhans, and D. Haussler. A discriminative framework for detecting remote protein homologies. *Journal of Computational Biology*, 7(1-2):95–114, 2000.

[16] T. Jebara. *Machine Learning: Discriminative and Generative*, volume 755 of *The Springer International Series in Engineering and Computer Science*. Springer, 2004.

[17] F. Jelinek and R.L. Mercer. Interpolated estimation of Markov source parameters from sparse data. In *Proc. Workshop on Pattern Recognition in Practice*, pages 381–397, 1980.

[18] D. Jurafsky and J.H. Martin. *Speech and Language Processing*. Prentice Hall, 2008.

[19] D. Klakow. Log-linear interpolation of language models. In *Proc. 5^{th} International Conference on Spoken Language Processing*, pages 1695–1698, 1998.

[20] H.R. Lee and J.S.R. Jang. i-Ring: A system for humming transcription and chord generation. In *Proc. IEEE International Conference on Multimedia and Expo (ICME)*, volume 2, pages 1031–1034, 2004.

[21] F.J. Och and H. Ney. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 295–302, 2002.

[22] F. Pachet and P. Roy. Musical harmonization with constraints: A survey. *Constraints*, 6(1):7–19, 2001.

[23] J.F. Paiement, D. Eck, and S. Bengio. Probabilistic melodic harmonization. In *Proc. 19th Canadian Conf. on Artificial Intelligence*, pages 218–229, 2006.

[24] H. Papadopoulos and G. Peeters. Large-scale study of chord estimation algorithms based on chroma representation and hmm. In *Proc. IEEE Int. Workshop on Content-Based Multimedia Indexing (CBMI)*, pages 53–60. IEEE, 2007.

[25] S. Phon-Amnuaisuk and G. Wiggins. The four-part harmonisation problem: a comparison between genetic algorithms and a rule-based system. In *Proc. Artificial Intelligence and Simulation of Behavior conference*, volume 99, pages 28–34, 1999.

[26] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. ISBN 3-900051-07-0.

[27] C. Raphael and J. Stoddard. Functional harmonic analysis using probabilistic models. *Computer Music Journal*, 28(3):45–52, 2004.

[28] C. Rathinavelu and L. Deng. The trended HMM with discriminative training for phonetic classification. In *Proc. 4^{th} International Conference on Spoken Language (ICSLP)*, volume 2, pages 1049–1052, 1996.

[29] J. Rennie and R. Rifkin. Improving multiclass text classification with the support vector machine. Technical Report AIM-2001-026, Massachusetts Institute of Technology, 2001.

[30] R. Scholz, E. Vincent, and F. Bimbot. Robust modeling of musical chord sequences using probabilistic n-grams. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 53–56, 2009.

[31] I. Simon, D. Morris, and S. Basu. MySong: automatic accompaniment generation for vocal melodies. In *Proc. 26th SIGCHI Conference on Human Factors in Computing Systems*, pages 725–734, 2008.

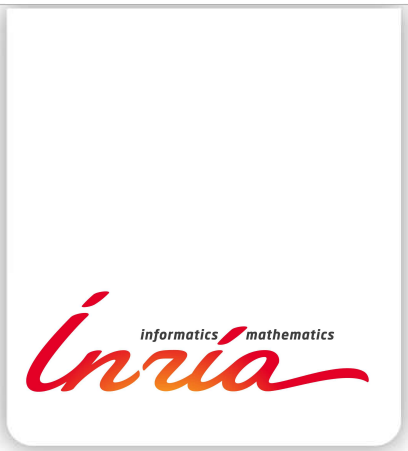
[32] L. Steels. Learning the craft of musical composition. In *Proc. International Computer Music Conference (ICMC)*, pages A–27–A–31, 1986.

[33] D. Temperley and D. Sleator. Modeling meter and harmony: A preference-rule approach. *Computer Music Journal*, 23(1):10–27, 1999.

[34] D. Temperley and D. Sleator. Harmonic Analyzer. `http://www.cs.cmu.edu/~sleator/harmonic-analysis/`, August 2012.

[35] P.C. Woodland and D. Povey. Large scale discriminative training of hidden Markov models for speech recognition. *Computer Speech & Language*, 16(1):25–47, 2002.

[36] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems*, 22(2):179–214, 2004.



RESEARCH CENTRE
RENNES – BRETAGNE ATLANTIQUE

Campus universitaire de Beaulieu
35042 Rennes Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399